



“Proyecto Academia Musical”

Taller de conceptos y principios de POO

Aprendiz:

Manuel Prudencio Pertuz Pérez

Servicio Nacional de Enseñanza – SENA

Centro Minero Regional Boyacá.

2977343 – Análisis Y Desarrollo de Software.

Instructor:

Luis Edilberto Díaz Sandoval

Julio - 2025.



Tabla de Contenido

1. Introducción	3
2. Glosario de terminología en POO	4
4. Conclusiones	6
5. Bibliografía	6



1. Introducción

En el desarrollo de software, especialmente cuando trabajamos con sistemas complejos como el de nuestra "Jacquin Academia Musical", resulta fundamental contar con herramientas conceptuales que nos ayuden a representar, modelar y construir el sistema de manera organizada y eficiente. La Programación Orientada a Objetos (POO) es uno de los paradigmas más potentes para ello. Este taller tiene como objetivo revisar y comprender los conceptos básicos de la POO a través de un glosario explicativo, además de una reflexión personal sobre sus pilares fundamentales, explicados con un lenguaje claro, cercano y adaptado a nuestra manera de aprender paso a paso.



2. Glosario de terminología en POO

POO (Programación Orientada a Objetos): Es una forma de programar que se basa en representar cosas del mundo real como "objetos" dentro del software. Cada objeto tiene datos (atributos) y acciones (métodos) que puede realizar.

Clase: Es como el molde o plano de una cosa. Por ejemplo, si vamos a representar estudiantes, la clase "Estudiante" define qué datos y comportamientos tienen todos los estudiantes.

Objeto: Es una instancia real de una clase. Siguiendo el ejemplo anterior, un estudiante llamado Juan sería un objeto de la clase Estudiante.

Atributos: Son las características o propiedades que tiene un objeto. Por ejemplo, nombre, edad, instrumento musical que estudia.

Métodos: Son las acciones o comportamientos que un objeto puede realizar. Como inscribirse a un curso o consultar su progreso.

Herencia: Permite que una clase (por ejemplo, Profesor) herede las características de otra clase más general (como Persona). Esto evita repetir código.

Eventos: Son acciones que ocurren dentro del sistema y que un objeto puede "escuchar" o responder. Por ejemplo, cuando un estudiante finaliza un curso, se puede activar un evento para notificar al administrador.

Abstracción: Es enfocarse solo en lo necesario. Es decir, cuando creamos un objeto, mostramos solo lo relevante para quien lo usa, ocultando los detalles internos.

Encapsulamiento: Es proteger los datos del objeto, permitiendo acceder o modificarlos solo a través de métodos especiales. Como cuando un estudiante solo puede cambiar su correo desde una función controlada.



Polimorfismo: Significa que una misma acción puede funcionar diferente según el objeto que la ejecuta. Por ejemplo, un método "evaluar()" puede hacer una evaluación teórica o práctica, según el curso.

3. Características y pilares básicos de la poo

Para mí, la Programación Orientada a Objetos tiene una gran ventaja: me permite imaginar que estoy organizando mi código como si fuera una pequeña ciudad donde cada objeto es un habitante con su propia función, nombre, responsabilidades y límites.

Los cuatro pilares que le dan vida a esta forma de programar son:

- **Abstracción:** Gracias a ella, podemos crear representaciones simples de cosas complejas. Por ejemplo, en nuestra academia musical, no necesito saber cómo se calcula exactamente el pago de un curso, solo que existe una función que lo hace.
- **Encapsulamiento:** Esto me ayuda a proteger los datos importantes. Si un estudiante tiene una nota, esta no puede ser modificada de cualquier forma, solo desde una función que garantice que sea correcto.
- **Herencia:** Me permite ahorrar tiempo y escribir menos. Si ya definí lo que es una Persona, puedo crear Estudiantes y Profesores que tomen esas características básicas y añadan las suyas propias.
- **Polimorfismo:** Este pilar me parece genial porque me permite usar el mismo nombre para acciones que pueden cambiar según el contexto. Así, el código es más flexible y adaptado.



4. Conclusiones

La Programación Orientada a Objetos es mucho más que una técnica de programación; es una forma ordenada y natural de pensar soluciones. Nos permite acercarnos a los problemas del mundo real desde una perspectiva organizada, reutilizable y flexible. Gracias a conceptos como clases, objetos y los pilares fundamentales, podemos construir sistemas como el de "Jacquin Academia Musical" con claridad, seguridad y facilidad de mantenimiento. El aprendizaje paso a paso, como hemos trabajado hasta ahora, hace que estos conceptos cobren vida de forma significativa.

5. Bibliografía

- Booch, G. et al. (2007). *Object-Oriented Analysis and Design with Applications*.
- Aguilar, J. (1998, 2021). *Principios de la POO*.
- Rumbaugh, J. et al. (2004). *The Unified Modeling Language Reference Manual*.
- Pressman, R. (1998). *Ingeniería del software*.
- Cuevas, F. (1991). *Técnicas y herramientas CASE*.
- ITCA (2021). *UML para el modelado estático en software*.
- EcuRed (2021). *Patrones de diseño de software*.